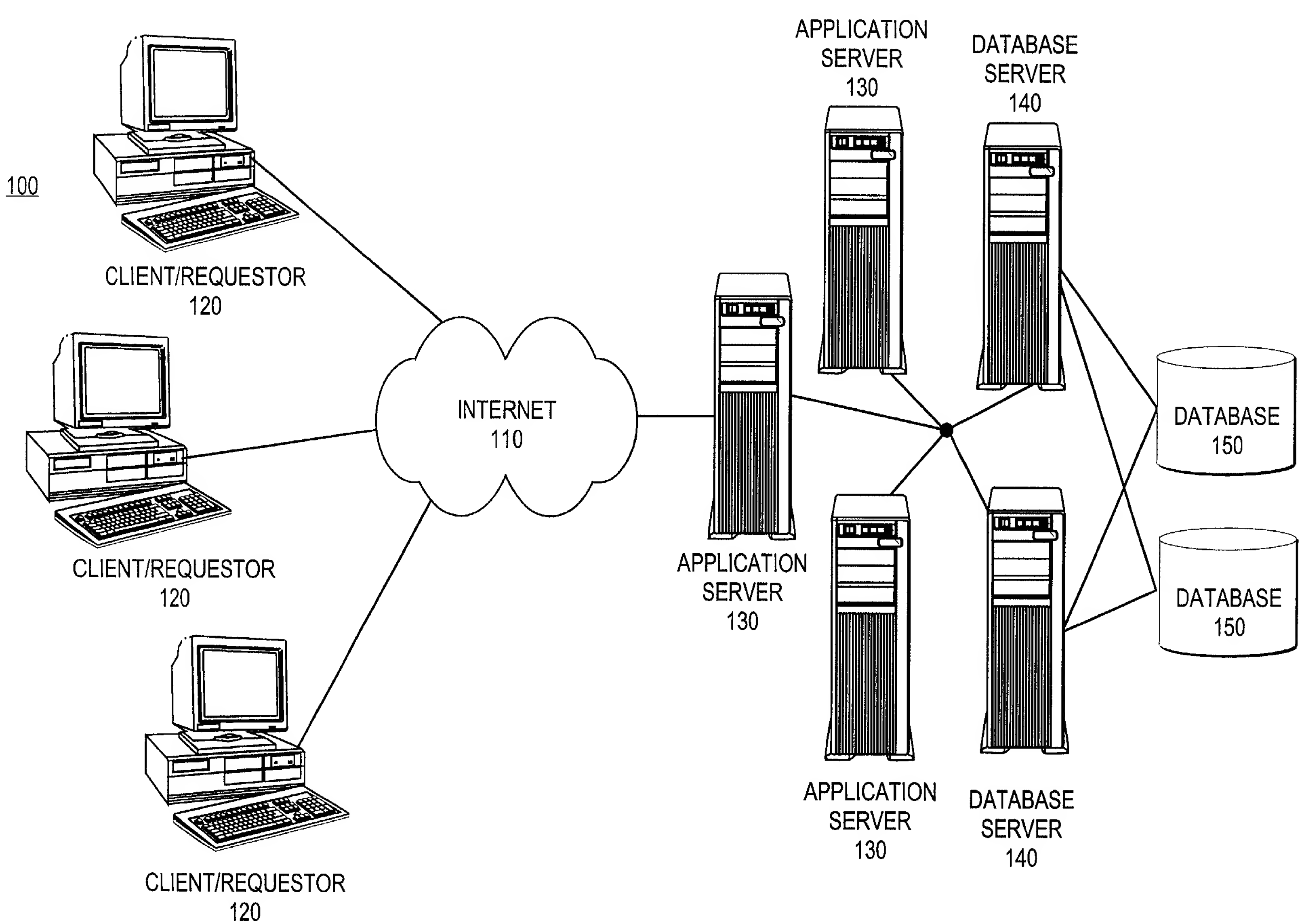
100
CLIENT/REQUESTOR
120
CLIENT/REQUESTOR
120
CLIENT/REQUESTOR
120
INTERNET
110
APPLICATION
SERVER
130
APPLICATION
SERVER
130
APPLICATION
SERVER
130
DATABASE
SERVER
140
DATABASE
SERVER
140
DATABASE
150
DATABASE
150

FIG. 1

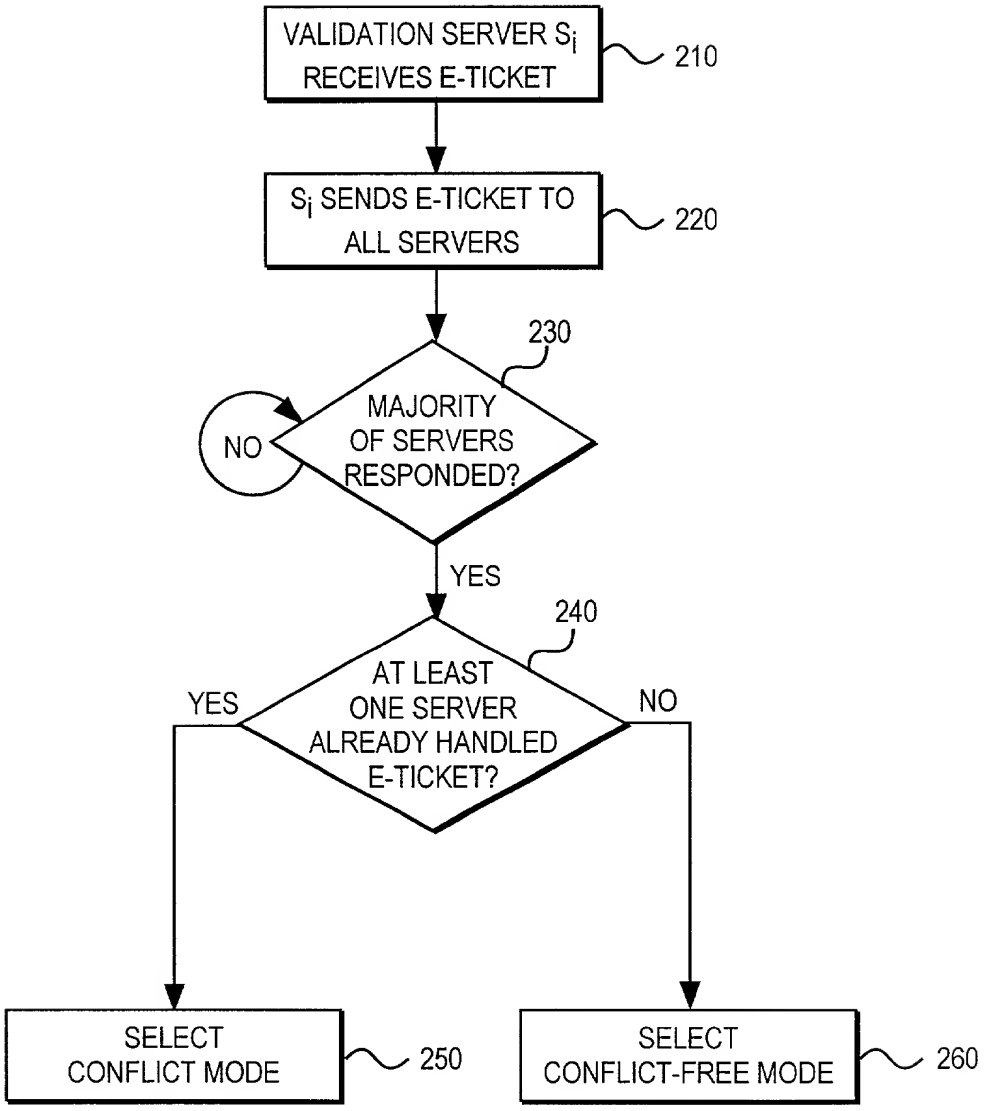
VALIDATION SERVER S_i RECEIVES E-TICKET
210
S_i SENDS E-TICKET TO ALL SERVERS
220
230
MAJORITY OF SERVERS RESPONDED?
NO
YES
240
AT LEAST ONE SERVER ALREADY HANDLED E-TICKET?
YES
NO
SELECT CONFLICT MODE
250
SELECT CONFLICT-FREE MODE
260

FIG. 2

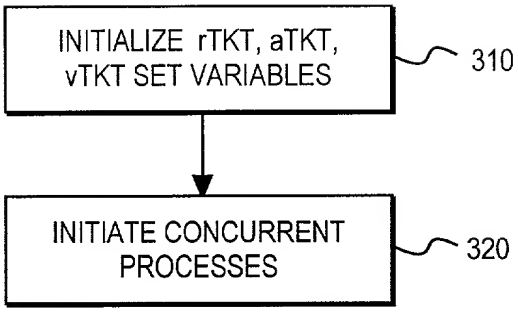
INITIALIZE rTKT, aTKT, vTKT SET VARIABLES
310
INITIATE CONCURRENT PROCESSES
320

FIG. 3

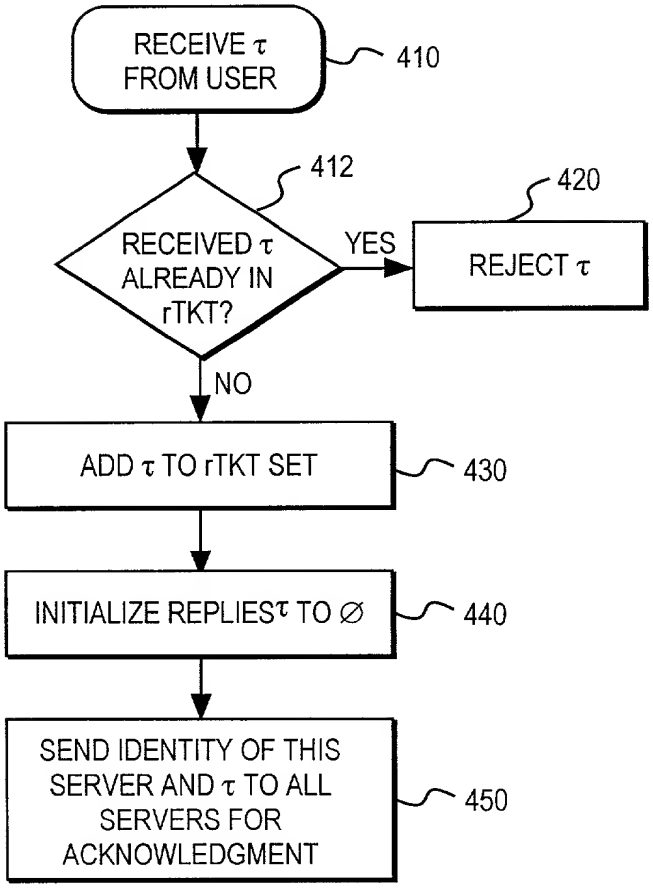
RECEIVE τ FROM USER
410
412
RECEIVED τ ALREADY IN rTKT?
YES
420
REJECT τ
NO
ADD τ TO rTKT SET
430
INITIALIZE REPLIES$^{\tau}$ TO ∅
440
SEND IDENTITY OF THIS SERVER AND τ TO ALL SERVERS FOR ACKNOWLEDGMENT
450

FIG. 4

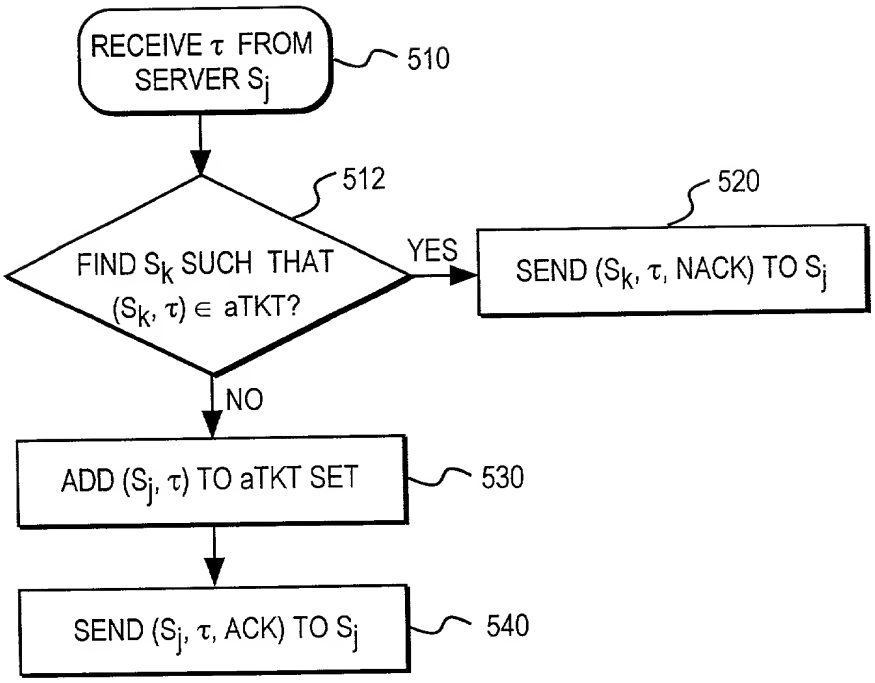
RECEIVE τ FROM SERVER S_j
510
512
FIND S_k SUCH THAT (S_k, τ) ∈ aTKT?
YES
520
SEND (S_k, τ, NACK) TO S_j
NO
ADD (S_j, τ) TO aTKT SET
530
SEND (S_j, τ, ACK) TO S_j
540

FIG. 5

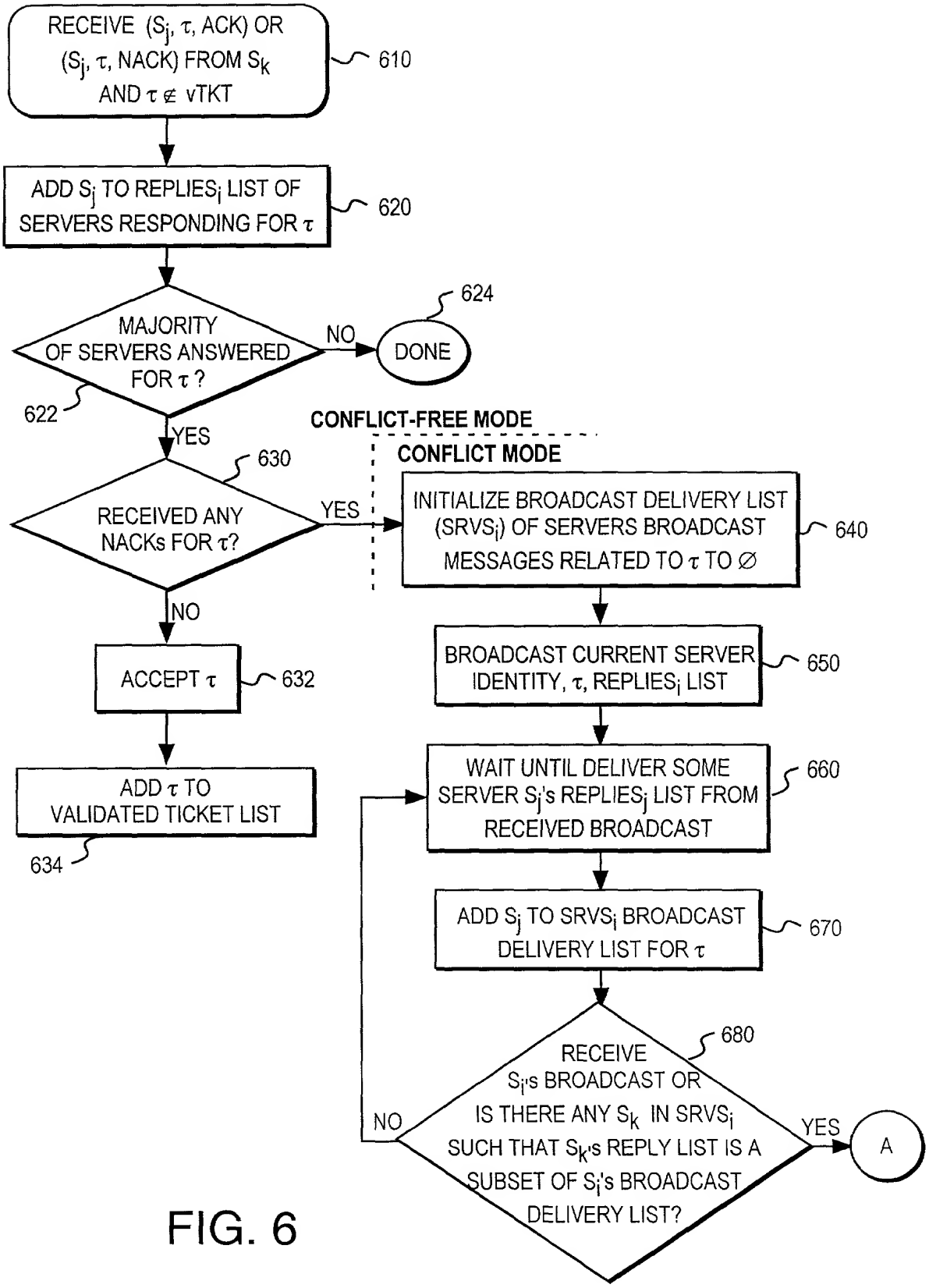
RECEIVE (S_j, τ, ACK) OR (S_j, τ, NACK) FROM S_k AND τ ∉ vTKT
610
ADD S_j TO $REPLIES_i$ LIST OF SERVERS RESPONDING FOR τ
620
MAJORITY OF SERVERS ANSWERED FOR τ ?
622
NO
624
DONE
YES
CONFLICT-FREE MODE
CONFLICT MODE
RECEIVED ANY NACKs FOR τ?
630
YES
INITIALIZE BROADCAST DELIVERY LIST ($SRVS_i$) OF SERVERS BROADCAST MESSAGES RELATED TO τ TO ∅
640
NO
ACCEPT τ
632
BROADCAST CURRENT SERVER IDENTITY, τ, $REPLIES_i$ LIST
650
ADD τ TO VALIDATED TICKET LIST
634
WAIT UNTIL DELIVER SOME SERVER S_j's $REPLIES_j$ LIST FROM RECEIVED BROADCAST
660
ADD S_j TO $SRVS_i$ BROADCAST DELIVERY LIST FOR τ
670
RECEIVE S_i's BROADCAST OR IS THERE ANY S_k IN $SRVS_i$ SUCH THAT S_k's REPLY LIST IS A SUBSET OF S_i's BROADCAST DELIVERY LIST?
680
NO
YES
A

FIG. 6

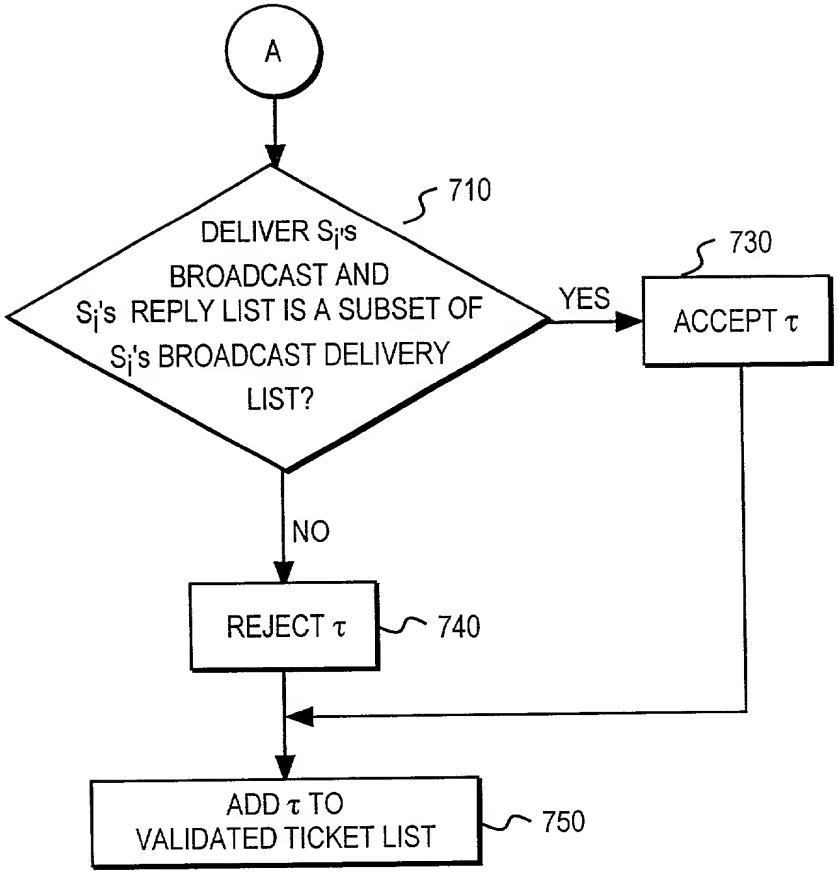
A
710
DELIVER S_i's BROADCAST AND S_i's REPLY LIST IS A SUBSET OF S_i's BROADCAST DELIVERY LIST?
YES
730
ACCEPT τ
NO
REJECT τ
740
ADD τ TO VALIDATED TICKET LIST
750

FIG. 7

1: **'INITIALIZATION**
2: $rTKT_i \leftarrow \varnothing$
3: $vTKT_i \leftarrow \varnothing$
4: $aTKT_i \leftarrow \varnothing$
5: **'IDENTIFY MODE; HANDLE IF CONFLICT-FREE MODE**
6: **WHEN** RECEIVE τ FROM USER
7: **IF** $\tau \in rTKT_i$ **THEN**
8: REJECT(τ)
9: **ELSE**
10: $rTKT_i \leftarrow rTKT_i \cup \{\tau\}$
11: $REPLIES_i^{\tau} \leftarrow \varnothing$
12: SEND(S_i, τ, NEWTKT) TO ALL
13: **WHEN** RECEIVE (S_j, τ_j, NEWTKT) FROM S_j
14: **IF** $\left(\exists\ S_k \ni (S_k, \tau_j) \in aTKT_i\right)$ **THEN**
15: SEND(S_k, τ_j, NACK) TO S_j
16: **ELSE**
17: $aTKT_i \leftarrow aTKT_i \cup \left\{(S_j, \tau_j)\right\}$
18: SEND(S_j, τ_j, ACK) TO S_j
19: **WHEN** (RECEIVE(S_j, τ_j, ACK) OR (S_j, τ_j, NACK) FROM S_k) AND ($\tau_j \notin vTKT_i$) **THEN**
20: $REPLIES_i^{\tau_j} \leftarrow REPLIES_i^{\tau_j} \cup \left\{S_j\right\}$
21: **IF** COUNT($REPLIES_i^{\tau_j}$) $\geq \lceil (N+1)/2 \rceil$ **THEN**
22: **IF** (RECEIVED(*, τ_j, ACK) $\forall S_k \in REPLIES_i^{\tau_j}$ **THEN**
23: ACCEPT(τ_j)
24: **ELSE**
25: **'CONFLICT MODE**
26: $SRVS_i^{\tau_j} \leftarrow \varnothing$
27: BROADCAST(S_i, τ_j, $REPLIES_i^{\tau_j}$)
28: **REPEAT**
29: **WAIT UNTIL** DELIVER(S_j, τ_j, $REPLIES_j^{\tau_j}$)
30: $SRVS_i^{\tau_j} \leftarrow SRVS_i^{\tau_j} \cup \left\{S_j\right\}$
31 **UNTIL** ($j = i$ **OR** $\exists S_k \in SRVS_i^{\tau_j} \ni REPLIES_k^{\tau_j} \subseteq SRVS_i^{\tau_j}$)
32: **IF** (DELIVERED((S_i, τ_j, $REPLIES_i^{\tau_j}$)) **AND** $REPLIES_i^{\tau_j} \subseteq SRVS_i^{\tau_j}$ **THEN**
33: ACCEPT(τ_j)
34: **ELSE**
35: REJECT(τ_j)
36: $vTKT_i \leftarrow vTKT_i \cup \left\{\tau_j\right\}$

FIG. 8